What is claimed is:

1. A method of processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the method comprising:

(a) sorting at least a subset of a plurality of access requests directed to the DASD based upon the requesters associated therewith to generate a first ordered set of access requests;

(b) sorting at least a subset of the access requests in the first ordered set of access requests based upon the positions associated therewith to generate a second ordered set of access requests; and

(c) issuing each of the access requests in the second ordered set of access requests in sequence to the DASD.

2. The method of claim 1, further comprising moving access requests between a first queue and a second queue, wherein sorting the access requests to generate the first ordered set of access requests includes receiving inbound access requests into the first queue, and sorting the access requests received into the first queue based upon the requesters associated therewith, wherein sorting the access requests in the first ordered set includes sorting access requests moved into the second queue based upon the positions associated therewith, and wherein issuing the access requests in the second ordered set includes removing issued access requests from the second queue.

3. The method of claim 2, wherein sorting the access requests received into the first queue includes, for each inbound access request:

(a) determining whether another access request in the first queue is associated with the same requester as the inbound access request;

(b) if so, storing the inbound access request after a last access request associated with the same requester as the inbound access request; and

(c) if not, arranging the inbound access request within the first queue based upon a requester identifier associated therewith.

4. The method of claim 3, wherein moving access requests between the first queue and the second queue is performed in response to determining that the second queue is empty.

5. The method of claim 4, wherein moving access requests between the first queue and the second queue includes, for each requester associated with an access request stored in the first queue, moving at least one access request associated with such requester into the second queue.

6. The method of claim 2, wherein moving access requests between the first queue and the second queue further includes moving a batch of access requests between the first queue and second queue, and wherein sorting the access requests moved into the second queue includes reversing a sort order for each successive batch of access requests.

7. The method of claim 1, wherein sorting the access requests in the first ordered set includes alternately sorting access requests in ascending and descending order.

8. The method of claim 1, wherein each access request is associated with one of a plurality of requesters in a computer coupled to the DASD, with each requester being a computer task executing on the computer.

9. A method of processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the method comprising:

(a) sorting a plurality of access requests directed to the DASD based upon both the requesters and the positions associated therewith; and

(b) issuing the sorted access requests to the DASD.

10. The method of claim 9, wherein sorting the plurality of access requests includes:

(a) sorting at least a subset of a plurality of access requests based upon the requesters associated therewith to generate a first ordered set of access requests; and

(b) sorting at least a subset of the access requests in the first ordered set of access requests based upon the positions associated therewith to generate a second ordered set of access requests, wherein issuing the sorted access requests includes issuing each of the access requests in the second ordered set of access requests in sequence to the DASD.

11. The method of claim 9, wherein sorting the plurality of access requests includes:

(a) sorting at least a subset of a plurality of access requests using one of a fair scheduling algorithm and an elevator scheduling algorithm to generate a first ordered set of access requests; and

(b) sorting at least a subset of the access requests in the first ordered set of access requests using the other of the fair scheduling algorithm and the elevator scheduling algorithm.

12. A method of processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the method comprising:

(a) receiving incoming access requests into a first queue that is sorted based upon the requester associated with each access request stored in the first queue;

(b) moving at least one access request from the first queue to a second queue that is sorted based upon the position associated with each access request stored in the second queue; and

(c) sequentially issuing access requests from the second queue to the DASD.

13. An apparatus for use in processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the apparatus comprising:

(a) a memory; and

(b) a program, resident in the memory, the program configured to sort a plurality of access requests directed to the DASD based upon both the requesters and the positions associated therewith, and issue the sorted access requests to the DASD.

14. The apparatus of claim 13, wherein the program is configured to sort the plurality of access requests by sorting at least a subset of a plurality of access requests using one of a fair scheduling algorithm and an elevator scheduling algorithm to generate a first ordered set of access requests; and sorting at least a subset of the access requests in the first ordered set of access requests using the other of the fair scheduling algorithm and the elevator scheduling algorithm.

15. The apparatus of claim 13, wherein the program is configured to sort the plurality of access requests by sorting at least a subset of a plurality of access requests based upon the requesters associated therewith to generate a first ordered set of access requests, and sorting at least a subset of the access requests in the first ordered set of access requests based upon the positions associated therewith to generate a second ordered set of access requests, and wherein the program is configured to issue the sorted access requests by issuing each of the access requests in the second ordered set of access requests in sequence to the DASD.

16. The apparatus of claim 15, wherein the program is further configured to:

(a) move access requests between a first queue and a second queue;

(b) sort the access requests to generate the first ordered set of access requests by receiving inbound access requests into the first queue and sorting the access requests received into the first queue based upon the requesters associated therewith;

(c) sort the access requests in the first ordered set by sorting access requests moved into the second queue based upon the positions associated therewith; and

(d) issue the access requests in the second ordered set by removing issued access requests from the second queue.

17. The apparatus of claim 16, wherein the program is configured to sort the access requests received into the first queue by, for each inbound access request:

(a) determining whether another access request in the first queue is associated with the same requester as the inbound access request;

(b) if so, storing the inbound access request after a last access request associated with the same requester as the inbound access request; and

(c) if not, arranging the inbound access request within the first queue based upon a requester identifier associated therewith.

18. The apparatus of claim 17, wherein the program is configured to move access requests between the first queue and the second queue in response to determining that the second queue is empty.

19. The apparatus of claim 18, wherein the program is configured to move access requests between the first queue and the second queue by, for each requester associated with an access request stored in the first queue, moving at least one access request associated with such requester into the second queue.

20. The apparatus of claim 16, wherein the program is configured to move access requests between the first queue and the second queue by moving a batch of access requests between the first queue and second queue, and wherein the program is configured to sort the access requests moved into the second queue by reversing a sort order for each successive batch of access requests.

21. The apparatus of claim 15, wherein the program is configured to sort the access requests in the first ordered set by alternately sorting access requests in ascending and descending order.

22. The apparatus of claim 15, further comprising a plurality of requester computer tasks resident in the memory, wherein each access request is associated with one of the plurality of requester computer tasks.

23. An apparatus for use in processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the apparatus comprising:

(a) a first queue configured to receive incoming access requests, the access requests in the first queue sorted based upon the requesters associated therewith;

(b) a second queue configured to issue outgoing access requests to the DASD, the access requests in the second queue sorted based upon the positions associated therewith; and

(c) control logic coupled to the first and second queues and configured to selectively move access requests from the first queue to the second queue.

24. The apparatus of claim 23, further comprising a memory; and a processor coupled to the memory, wherein the control logic comprises request processing program code resident in the memory.

25. The apparatus of claim 24, further comprising an operating system resident in the memory, wherein the request processing program code and the first and second queues are resident in the operating system.

26. The apparatus of claim 25, wherein the operating system includes a DASD hardware driver that interfaces the apparatus with the DASD, and wherein the request processing program code and the first and second queues are resident in the DASD hardware driver.

27. The apparatus of claim 24, wherein the first and second queues are resident in the memory.

28. A program product, comprising:

(a) a program for use in processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the program configured to sort a plurality of access requests directed to the DASD based upon both the requesters and the positions associated therewith, and issue the sorted access requests to the DASD; and

(b) a signal bearing medium bearing the program.

29. The program product of claim 28, wherein the signal bearing medium includes at least one of a recordable medium and a transmission type medium.